IOSR Journal of Engineering (IOSRJEN)
e-ISSN: 2250-3021, p-ISSN: 2278-8719, www.iosrjen.org
Volume 2, Issue 9 (September 2012), PP 06-10

Performance factors of cloud computing data centers using M/G/m/m+r queuing systems

Prof. Bharathi M, Prof. Sandeep Kumar P, Prof. Poornima G V
[1]Dept. of CSE, SJCIT, VTU. [2]Dept. of CSE, EPCEW, VTU. [3]Dept. of CSE, EPCEW, VTU

Abstract— *Cloud computing is a novel paradigm for the provision of computing infrastructure, which aims to shift the location of the computing infrastructure to the network in order to reduce the costs of management and maintenance of hardware and software resources. Cloud computing systems fundamentally provide access to large amounts of data. Resources provided by cloud computing systems hide a great deal of information from the user through virtualization. In this paper, the cloud data center is modelled as M/G/m/m+r queuing system with a single task arrivals and a task request buffer of finite capacity.*

Keywords—*Cloud computing, performance analysis, response time, queuing theory, markov chain process.*

I. INTRODUCTION

Cloud computing is a paradigm shift from the client server model. Cloud computing allows users in an organization to access shared resources, applications, servers, computers, etc. over the internet on demand from a service provider. The provider can calculate the time bound resource utilization as per the service level agreement and bill the organization based on the usage. This is just like using the electricity from your provider and paying the electric bill based on the usage as captured by the meter. This will help the companies to optimize their utilization of IT assets thereby cutting down on any unnecessary IT costs. So this way organization can leverage on the cloud computing model to reduce the capital expenditure on IT.

The main advantage of having multiple servers in Cloud computing is, the system performance increases effectively by reducing the mean queue length and waiting time than compared to the traditional approach of having only single server so that the CCU's need not wait for a long period of time and also queue length need not be large.

In this paper we model the cloud center as an M/G/m/m+r queueing system with single task arrivals and a task request buffer of finite capacity. We evaluate its performance using a movel analytical model and solve it to obtain important performance factors like mean number of tasks in the system.

The remainder of the paper is organized as follows. Section 2 gives a brief overview of existing work on various queuing models and assumptions. Section 3 discusses our analytical model in detail. We present and discuss analytical results in section 4. Our findings are summarized in Section 5, where we also outline the directions for future work.

II. MODELS AND ASSUMPTIONS

The kendall's classification of queuing systems exists in several modifications.

Queuing models are generally constructed to represent the steady state of a queuing system, that is, the typical, long run or average state of the system. As a consequence, these are stochastic models that represent the probability that a queuing system will be found in a particular configuration or state.

A general procedure for constructing and analysing such queuing models is:

1. Identify the parameters of the system, such as the arrival rate, service time, queue capacity, and perhaps draw a diagram of the system.
2. Identify the system states. (A state will generally represent the integer number of customers, people, jobs, calls, messages, etc. in the system and may or may not be limited.)
3. Draw a state transition diagram that represents the possible system states and identify the rates to enter and leave each state. This diagram is a representation of a Markov chain.
4. Because the state transition diagram represents the steady state situation between state there is a balanced flow between states so the probabilities of being in adjacent states can be related mathematically in terms of the arrival and service rates and state probabilities.
5. Express all the state probabilities in terms of the empty state probability, using the inter-state transition relationships.
6. Determine the empty state probability by using the fact that all state probabilities always sum to 1.

M/M/1/∞/∞ represents a single server that has unlimited queue capacity and infinite calling population, both arrivals and service are Poisson (or random) processes, meaning the statistical distribution of both the inter-arrival times and the service times follow the exponential distribution. Because of the mathematical nature of the exponential distribution, a number of quite simple relationships can be derived for several performance measures based on knowing the arrival rate and service rate.

M/G/1/∞/∞ represents a single server that has unlimited queue capacity and infinite calling population, while the arrival is still Poisson process, meaning the statistical distribution of the inter-arrival times still follow the exponential distribution, the distribution of the service time does not. The distribution of the service time may follow any general statistical distribution, not just exponential. Relationships can still be derived for a (limited) number of performance measures if one knows the arrival rate and the mean and variance of the service rate. However the derivations are generally more complex and difficult.

In [5], the distribution of response time was obtained for a cloud center modelled as anM/M/m/m+r queueing system. Both inter-arrival and service times were assumed to be exponentially distributed, and the system had a finite buffer of size m+r. The response time was broken down into waiting, service, and execution periods, assuming that all three periods are independent which is unrealistic, according to authors' own argument.

Analysis in the case where inter-arrival time and/or service time are not exponential is more complex. Most theoretical analyses have relied on extensive research in performance evaluation of M/G/m queueing systems [6], [7], [8]. However, the probability distributions of response time and queue length in M/G/m and M/G/m/m+r cannot be obtained in closed form, which necessitated the search for a suitable approximation.

An approximate solution for steady-state queue length distribution in an M/G/m system with finite waiting space was described in [9]. As the approximation was given in an explicit form, its numerical computation is easier than when using earlier approximations [10], [11]. The proposed approach is exact for M/G/m/m+r when r=0, and reasonably accurate in the general case when r=0, but only when the number of servers m is small, below 10 or so.

Cloud centers differ from traditional queuing systems in a number of important aspects:
1. A cloud center can have a large number of facility (server) nodes, typically of the order of hundreds or thousands[2]; traditional queuing analysis rarely considers systems of this size.
2. Task service times must be modeled by a general, rather than the more convenient exponential, probability distribution. Moreover, the coefficient of variation of task service time may be high – i.e., well over the value of one.
3. Due to the dynamic nature of cloud environments, diversity of user's requests and time dependency of load, cloud centers must provide expected quality of service at widely varying loads[2][3].

While each of these aspects has been addressed in existing research, there is virtually no work that addresses all of them simultaneously. As most of these results rely on some approximation(s) to obtain a closed-form solution, they are not universally applicable
1. Approximations are reasonably accurate only when the number of servers is comparatively small, typically below 10 or so, which makes them unsuitable for performance analysis of cloud computing data centers.
2. Approximations are very sensitive to the probability distribution of task service times, and they become increasingly inaccurate when the coefficient of variation of the service time, CoV, increases toward and above the value of one.
3. Finally, approximation errors are particularly pronounced when the traffic intensity ρ is small, and/or when both the number of servers m and the CoV of the service time, are large.

As a result, the results mentioned above are not directly applicable to performance analysis of cloud computing server farms where one or more of the following holds: the number of servers is huge; the distribution of service times is unknown and does not, in general, follow any of the well-behaved probability distributions such as exponential distribution; finally, the traffic intensity can vary in an extremely wide range.

III. THE PROPOSED MODEL

There are three basic steps in the performance analysis process: data collection, data transformation, and data visualization. Data collection is the process by which data about program performance are obtained from an executing program. Data are normally collected in a file, either during or after execution, although in some situations it may be presented to the user in real time. Three basic data collection techniques can be distinguished:

1. Profiles record the amount of time spent in different parts of a program. This information, though minimal, is often invaluable for highlighting performance problems. Profiles typically are gathered automatically.

2. Counters record either frequencies of events or cumulative times. The insertion of counters may require some programmer intervention.
3. Event traces record each occurrence of various specified events, thus typically producing a large amount of data. Traces can be produced either automatically or with programmer intervention.

The raw data produced by profiles, counters, or traces are rarely in the form required to answer performance questions. Hence, data transformations are applied, often with the goal of reducing total data volume. Transformations can be used to determine mean values or other higher-order statistics or to extract profile and counter data from traces. For example, a profile recording the time spent in each subroutine on each processor might be transformed to determine the mean time spent in each subroutine on each processor, and the standard deviation from this mean. Similarly, a trace can be processed to produce a histogram giving the distribution of message sizes. Each of the various performance tools described in subsequent sections incorporates some set of built-in transformations; more specialized transformation can also be coded by the programmer.

Parallel performance data are inherently multidimensional, consisting of execution times, communication costs, and so on, for multiple program components, on different processors, and for different problem sizes. Although data reduction techniques can be used in some situations to compress performance data to scalar values, it is often necessary to be able to explore the raw multidimensional data. As is well known in computational science and engineering, this process can benefit enormously from the use of data visualization techniques. Both conventional and more specialized display techniques can be applied to performance data.

As we shall see, a wide variety of data collection, transformation, and visualization tools are available. When selecting a tool for a particular task, the issues like accuracy, simplicity, flexibility, intrusiveness, accuracy performance should be considered.

The proposed system models a cloud server farm as a *M/G/m/m + r* queuing system which indicates that the inter-arrival time of requests is exponentially distributed, while task service times are independent and identically distributed random variables that follow a general distribution with mean value of μ. The system under consideration contains *m* servers which render service in order of task request arrivals (FCFS). The capacity of system is *m + r* which means the buffer size for incoming request is equal to *r*. As the population size of a typical cloud center is relatively high while the probability that a given user will request service is relatively small, the arrival process can be modeled as a Markovian process.

An M/G/m/m+r queuing system may be considered as a semi-Markov process which can be analysed by exploiting the embedded Markov chain technique. Embedded Markov Chain technique requires selection of Markov points in which the state of the system is observed. Therefore we model the number of the tasks in the system (both those in service and those queued but not yet serviced) at the moments immediately before task request arrivals; if we enumerate these instances as 0, 1, 2, . . . , m + r, we obtain a homogeneous Markov chain. Therefore, the semi- Markov process records the state at arbitrary time while the embedded Markov chain only observes the state at which the system has an arrival. Task request arrivals follow a Poisson process, which means that task request inter-arrival time A is exponentially distributed with a rate of 1/λ. Finally, we assume that each task is serviced by a single server and we do not distinguish between installation (setup), actual task execution, and finalization components of the service time.

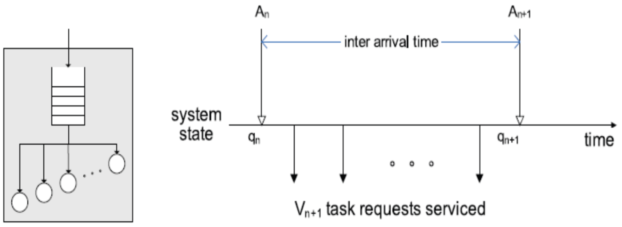

Figure 1: Embedded Markov points

The moments of task request arrivals are selected as Markov points. Two successive task request arrivals and task departures that (may) occur between them. The distribution of number of tasks in the system as well as the mean response time is calculated. The process then continues to estimate the transition probabilities associated with the embedded Markov chain. To find the elements of the transition probability matrix, we need to count the number of tasks departing from the system in between two successive arrivals. After finding the transition probability matrix P, we can establish the balance equations which link the probabilities of entering and leaving a state in equilibrium.

$$\pi_i = \sum_{j=0}^{m+r} \pi_j P_{ji}, 0 \le i \le m+r \qquad (1)$$

augmented by the normalization equation

$$\sum_{i=0}^{m+r} \pi_i = 1 \qquad (2)$$

Note that we have m + r + 2 equations which includes m + r + 1 linearly independent equations from (1) and one normalization equation from (2), but only m+ r +1 variables π_0, π_1, π_2, . . . , π_{m+r}. To obtain the unique equilibrium solution, we need to dispose of one of the equations. The best choice is to ignore the last equation in (1) since it carries less information about the system than the other ones. The balance equations will have a unique steady state solution if the corresponding Markov chain is ergodic. Once we obtain the steady state probabilities we are able to establish the probability generating functions (PGFs) for the number of tasks in the system at the time of a task arrival:

$$\Pi(z) = \sum_{k=0}^{m+r} \pi_z z^k \qquad (3)$$

For any Poisson arrivals system, PASTA property holds; thus, the PGF Π(z) for the distribution of the number of tasks in the system at the time of a task arrival is identical to the PGF P(z) for the distribution of the number of tasks in the system at an arbitrary time. Mean number of tasks in the system can be obtained. For M/G/m systems it is known that Q, the queue length, has the same distribution as W.

Therefore, the number of tasks which arrive during the waiting time is

$$Q(z) = W*(\lambda(1-z)) \qquad (4)$$

The left hand side of (4) can be calculated as

$$Q(z) = \sum_{k=0}^{m-1} \pi_k + \sum_{k=m}^{m+r} \pi_k z^{k-m} \qquad (5)$$

As we have a finite capacity system (i.e., there is blocking), let us define effective arrival rate as

$$\lambda_c = \lambda(1-\pi_{m+r})$$

Hence, we have

$$W*(s) = Q(z) \mid z = 1-(s/\lambda_c) = Q(1-s/\lambda_c) \qquad (6)$$

Moreover, the LST of response time is T*(s) = W*(s) B*(s), where B*(s) denotes the LST of service time. The i-th central moment, t(i), of the response time distribution is given by

$$t^{(i)} = \int_0^\infty x^i dT(x) = i\int_0^\infty x^{i-1}[1-T(x)]dx$$
$$= (-1)^i T*^{*}(0) \qquad i=1,2,3\ldots. \qquad (7)$$

IV. RESULTS

Table 1: System with m=100, queue size=1000 & arrival time=0.001

m/m+r	Utility	Queue	delay
100/100	0.182277	0.083077969	0.001165
100/200	0.182558	0.079785168	0.001167
100/300	0.183956	0.084594272	0.001176
100/400	0.182644	0.083568842	0.001175
100/500	0.181771	0.077923478	0.001134
100/600	0.184477	0.083892739	0.001214
100/700	0.182557	0.083274152	0.001179
100/800	0.181689	0.080762908	0.001185
100/900	0.183744	0.081713835	0.001165
100/1000	0.186199	0.091381943	0.001223

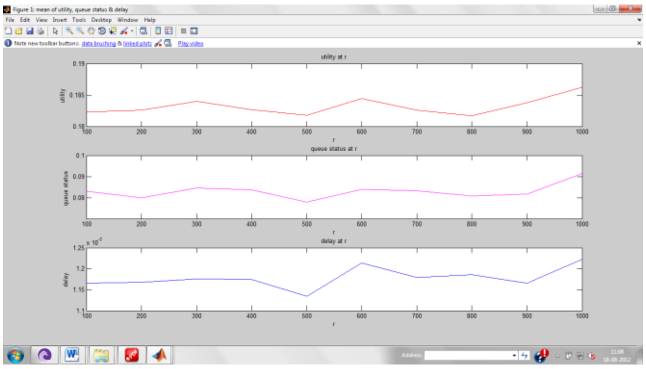

Figure 2: mean when m=100, queue size is 1000 & arrival time = 0.001

V. CONCLUSION

Performance evaluation of server farms is an important aspect of cloud computing which is of crucial interest for both cloud providers and cloud customers. In this project we have proposed an analytical model for performance evaluation of a cloud computing data centre.

In future, the results can be analysed using simulation. As mean is calculated skewness as well as standard deviation can be computed. The blocking probability and probability of immediate service can be computed

REFERENCES

[1] L. M. Vaquero, L. Rodero-Merino, J. Caceres, and M. Lindner, “A break in the clouds: towards a cloud definition,” *SIGCOMM Comput. Commun. Rev.*, vol. 39, pp. 50–55, Dec. 2008.

[2] Amazon Elastic Compute Cloud, *User Guide*, API Version ed., Amazon Web Service LLC or its affiliate, Aug. 2010. [Online]. Available: http://aws.amazon.com/documentation/ec2

[3] K. Xiong and H. Perros, “Service performance and analysis in cloud computing,” in *IEEE 2009 World Conference on Services*, Los Angeles, CA, 2009, pp. 693–700.

[4] J. Baker, C. Bond, J. Corbett, J. J. Furman, A. Khorlin, J. Larsonand, J. M. Leon, Y. Li, A. Lloyd, and V. Yushprakh, “Megastore: Providing Scalable, Highly Available Storage for Interactive Services,” in *Conference on Innovative Data Systems Research (CIDR)*, Jan. 2011, pp. 223–234.

[5] B. Yang, F. Tan, Y. Dai, and S. Guo, “Performance evaluation of cloud service considering fault recovery,” in *First Int’l Conference on Cloud Computing CloudCom 2009*, Beijing, China, Dec. 2009, pp. 571–576.

[6] B. N. W. Ma and J. W. Mark, “Approximation of the mean queue length of an M/G/cqueueing system,” *Operations Research*, vol. 43, pp. 158–165, 1998.

[7] M. Miyazawa, “Approximation of the queue-length distribution of an M/GI/s queue by the basic equations,” *Journal of Applied Probability*, vol. 23, pp. 443–458, 1986.

[8] D. D. Yao, “Refining the diffusion approximation for the M/G/m queue,” *Operations Research*, vol. 33, pp. 1266–1277, 1985.

[9] T. Kimura, “A transform-free approximation for the finite capacity M/G/s queue,” *Operations Research*, vol. 44, no. 6, pp. 984–988, 1996.

[10] P. Hokstad, “Approximations for theM/G/m queues,” *Operations Research*, vol. 26, pp. 510–523, 1978.

[11] H. C. Tijms, “Heuristics for finite-buffer queues,” *Probability in the Engineering and Informational Sciences*, vol. 6, pp. 277–285, 1992.